

# 爱创课堂前端培训

## CSS

第 7 天课堂笔记（本课程共 8 天）

班级：北京前端训练营 7 期

日期：2017 年 4 月 23 日

爱创课堂官网：[www.icketang.com](http://www.icketang.com)

# 目录

CSS.....	1
目录.....	2
复习.....	2
一、 固定定位.....	3
二、 z-index 属性.....	4
2.1 默认压盖顺序.....	4
2.2 自定义压盖顺序.....	4
2.3 z-index 应用.....	5
三、 浏览器的兼容性.....	6
四、 hack.....	6
4.1 html hack.....	6
4.2 css hack.....	7
4.2.1 css 值的 hack.....	7
4.2.2 css 选择器 Hack.....	8
五、 IE6 的兼容性.....	10
5.1 IE6 的选择器.....	10
5.2 盒模型的兼容问题.....	10
5.2.1 DTD 问题.....	10
5.2.2 最小高度.....	10
5.3 浮动的问题.....	11
5.3.1 浮动的盒子不会钻到底下.....	11
5.3.2 3 像素的 bug.....	11
5.3.3 overflow: hidden 失效.....	12
5.3 固定定位不支持.....	13

## 复习

元素脱离标注流的方法：浮动，绝对定位，固对定位。

相对定位：

position:relative;

四个方向的值：left, right; top,bottom;

相对定位没有脱离标准流；

保留原来的位置，新位置是相对于原位置的偏移。（形影分离）。

应用：①微调 ②做为绝对定位的参考盒子（子绝父相）

绝对定位：

绝对定位脱标。

没有祖先元素定位的参考盒子。

有 top 参与的情况：参考的是 **body** 左上角或者右上角；

有 bottom 参与的情况：参考的是**首屏**的左下角或者右下角；

针对祖先元素定位的参考盒子：

参考的是：**距离最近**，同时**有定位**的祖先元素；

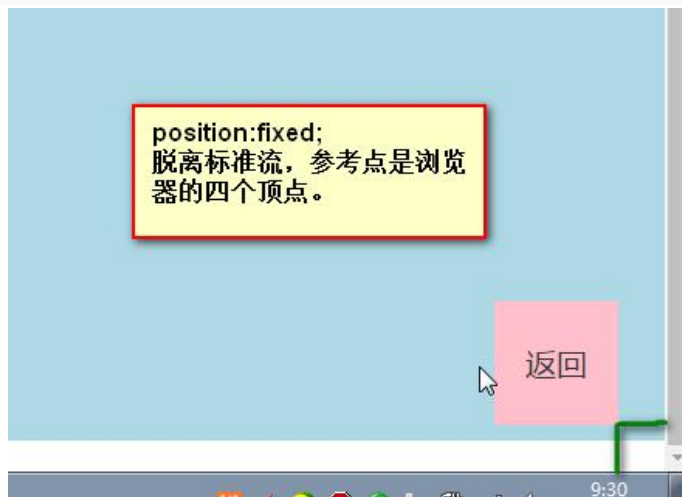
应用：轮播图；

## 一、固定定位

position:fixed;

四个方向值：left,right; top,bottom。

```
1 span{
2   width: 80px;
3   height: 80px;
4   line-height: 80px;
5   text-align: center;
6   font-size: 20px;
7   background: pink;
8   color:#333;
9   position:fixed;
10  right:30px;
11  bottom:30px;
12 }
```



固定定位脱离标准流；

参考点：**浏览器的四个顶点**。固定定位的盒子会始终出现在我们的浏览器窗口内。

应用：①返回顶部 ②固定导航栏

```
1 .nav{
2   width: 960px;
3   height:40px;
4   background: lightblue;
5   position:fixed;
6   top:0;
7   left:50%;
8   margin-left:-480px;
9 }
```



## 二、z-index 属性

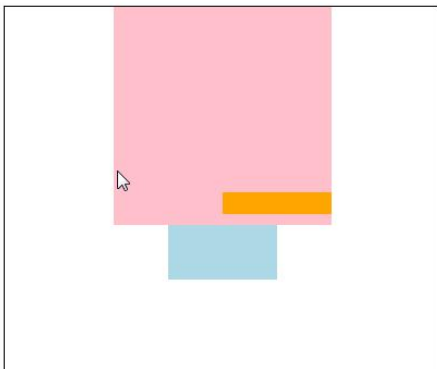
### 2.1 默认压盖顺序

- ①有定位的元素会压盖没有定位的元素。
- ②都有定位的元素，**html 结构中**在后面会压盖住在前面的。

```

1 <div class="box">
2   <div class="son3"></div>
3   <div class="son1"></div>
4   <div class="son2"></div>
5 </div>

```



### 2.2 自定义压盖顺序

自定义压盖顺序：z-index 属性

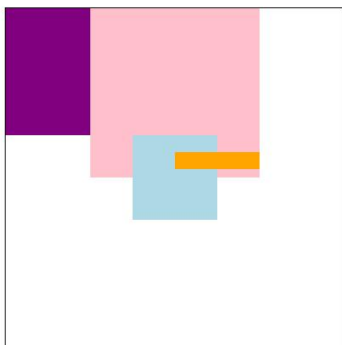
z-index 属性值：数字，没有单位。

- ①只有**定位的元素**可以加 z-index 属性，浮动的元素也不能加 z-index。

```

1 .son4{
2   width: 150px;
3   height: 150px;
4   background: purple;
5   float: left;
6   z-index:10;           //浮动的元素也不能加 z-index
7 }

```



- ②z-index 数值越大压盖顺序越靠上。

```

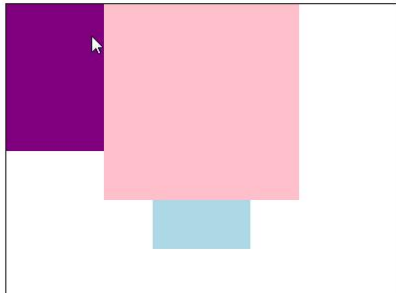
1 <div class="box">
2   <div class="son1"></div>           //z-index:6

```

```

3     <div class="son2"></div>      //z-index:5
4     <div class="son3"></div>      //z-index:4
5     <div class="son4"></div>
6 </div>

```

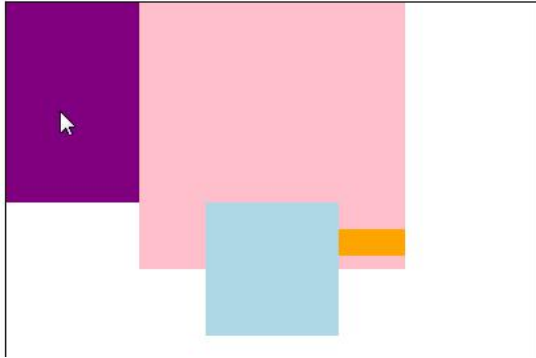


③z-index 数值相同时，html 结构中在后面会盖住在前面。

```

7 <div class="box">
8   <div class="son1"></div>      //z-index:4
9   <div class="son2"></div>      //z-index:4
10  <div class="son3"></div>      //z-index:4
11  <div class="son4"></div>
1 </div>

```

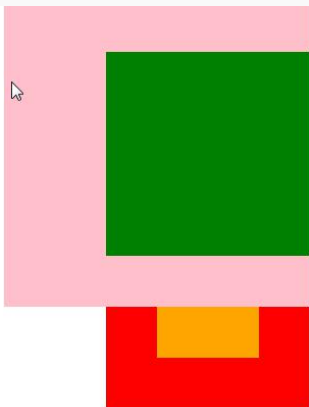


④父子都有 z-index 属性，父亲 z-index 数值小时，儿子数值再大没用。

```

1 <div class="big">                //z-index:3;
2   <div class="big_son"></div>    //z-index:2222;
3 </div>
4 <div class="small">              //z-index:2;
5   <div class="small_son"></div> //z-index:3333;
6 </div>

```



## 2.3 z-index 应用

轮播图

## 三、浏览器的兼容性

浏览器的五大厂商：IE 火狐 谷歌 苹果 欧朋

不同的浏览器有不同的版本；不同的版本有不同的兼容性。

IE 的兼容性要求：实际工作一般要求兼容到 IE8，IE6 的兼容性问题面试特别特别特别喜欢问。IE6 的兼容性处理的好，面试官会认为你的工作经验很长。

## 四、hack

hack 的分类：html hack 和 css hack

### 4.1 html hack

hacker:黑客

针对不同的浏览器在同一个 html 文件里，去写不同的 html 结构。

IE 浏览器有很多兼容性问题，给我们提供了一些接口。

```
<!--[if lte IE 9]>
    内容
<![endif]-->
```

表示 Hack 符里面的内容只在 IE9 及以下的浏览器内显示，IE10 及以上或者高级浏览器会认为这是注释。

if 是如果，lte 是 less than or equal，小于或者等于，IE 浏览器，9 代表版本。

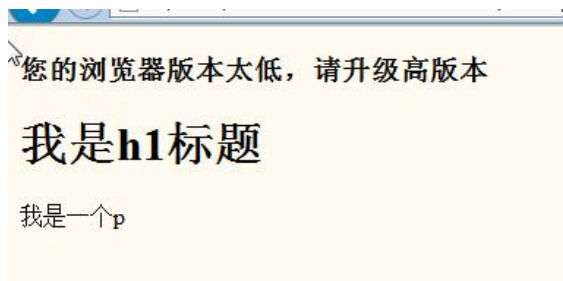
高级浏览器的显示效果

我是h1标题

我是一个p



低级浏览器的显示效果：（只有 IE9 及以下版本的浏览器可以显示）



lte:小于或者是等于该版本

lt:小于该版本

gt:大于该版本的显示

gte:大于等于该版本

html hack 表示低版本浏览器会认识这个 hack 能够正常渲染 hack，而高级浏览器会认为这个 hack 是注释，不在页面渲染。

针对 IE 6 有固定的一个 hack

```
1 <!--[if IE 6]>
2     <h3>这是 IE 浏览器才可以看见</h3>
3 <![endif]-->
```

表示只能 IE6 浏览器可以看到里面的内容，其他浏览器会认为是注释。

应用：给 IE6 添加特殊的一些样式或者 js 表示只有 IE 才可以加载。

```
1 <!--[if IE 6]>
2     <script type="text/javascript" src="js/png.js"></script>
3 <![endif]-->
4 <p>普通文字</p>
```

## 4.2 css hack

css hack 包括值 hack 和选择器的 hack。

### 4.2.1 css 值的 hack

*/\* IE 6 \*/*

```
.selector { _color: blue; }
.selector { -color: blue; }
```

Hack 符： *\_*、*-*

在属性名的前面加下划线或横线。

表示这个属性只有 IE6 认识，其他的都不认识这个属性。

```
1 .box{
2     width: 300px;
3     height: 300px;
4     background: pink;
5     color:#000;
6     -color:#fff;
7     font-size: 30px;
8 }
```



*/\* IE 6/7 \*/*

Hack 符： *! \$ & \* ( ) = % + @ , . / ` [ ] # ~ ? : < > |*

表示这个属性只有 IE6/7 认识，其他的都不认识这个属性。

```
1 .box{
2     width: 300px;
3     height: 300px;
```

```

4     background: pink;
5     color:#000;
6     /*这个属性只能 IE6/7 可以正常加载*/
7     !color:#fff;
8     font-size: 30px;
9 }

```

/\* IE 8/9 \*/

.selector { color: blue\0/; }

Hack 符: \0/。

需要写在属性值的后面。

只在 IE8、9 里认识。

```

1 .box{
2     width: 300px;
3     height: 300px;
4     background: pink;
5     color:#000;
6     /*这个属性只能 IE8/9 可以正常加载*/
7     font-size: 20px;
8     font-size:40px\0/;
9 }

```

/\* IE 6/7/8/9/10 \*/

.selector { color: blue\9; }

```

1 .box{
2     width: 300px;
3     height: 300px;
4     background: pink;
5     color:#000;
6     font-size: 20px;
7     /*这个属性只能 IE6/7/8/9/10 可以正常加载*/
8     background-color:lightblue\9;
9 }

```

Hack 符: \9.

需要写在属性值的后面，分号前。

着重记忆: IE6 的。

## 4.2.2 css 选择器 Hack

/\* IE 6 and below \*/

\*html.selector {}

这种选择器只在 IE6 里加载，其他的浏览器认为你的选择器是错的。

高级浏览器认为 html 已经是根标签，\*不是他的祖先元素。



```

1  * html .box{
2      width: 300px;
3      height: 300px;
4      background: lightblue;
5      color:#000;
6      font-size: 20px;
7  }

```

等价写法:

```

1  .box{
2      -width: 300px;
3      -height: 300px;
4      _color:#000;
5      _font-size: 20px;
6      -background-color:lightblue;
7  }

```

/\* IE 7 and below \*/

.selector, {}

这种选择器只在 IE7 及以下版本里加载，其他的浏览器认为你的选择器是错的。

```

1  .box,{
2      width: 300px;
3      height: 300px;
4      background: pink;
5      color:#000;
6      font-size: 20px;
7      background-color:lightblue\9;
8  }

```

等价写法:

```

1  .box{
2      !width: 300px;
3      !height: 300px;
4      !color:#000;
5      !font-size: 20px;
6      !background-color:lightblue;
7  }

```

/\* 除了 IE 6 \*/

html > body .selector {}

>子级选择器：只选择儿子级，后代其他级别不选。

IE6 不认识子级选择器。

除了 IE6 都能正常加载。

```

1  html>body .box{
2      width: 300px;
3      height: 300px;
4      background: pink;
5      color:#000;
6      font-size: 20px;
7  }

```

## 五、IE6 的兼容性

### 5.1 IE6 的选择器

之前的 7 种选择器：通配符，标签选择器，类选择器，id 选择器，后代选择器，交集选择器（div.box），并集选择器。

IE6 支持以上 7 种选择器，但是不支持交集选择器中连续类名的书写。

这种写法 IE6 不认识；

```
1      div.box.cl{
2          color:orange;
3          font-size: 30px;
4      }
```

解决办法：不写连续类名的选择器

```
1  div.box{}
```

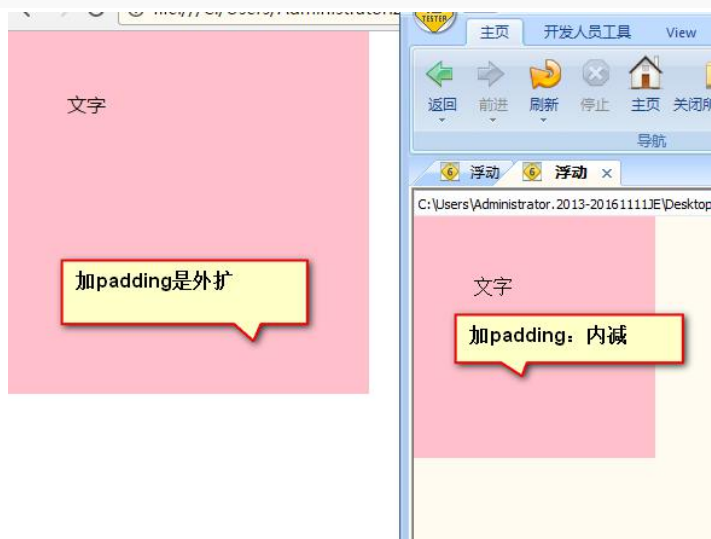
### 5.2 盒模型的兼容问题

#### 5.2.1 DTD 问题

如果 IE 不写 DTD 他的盒模型是内减的。

解决办法：必须 DTD

```
1
```



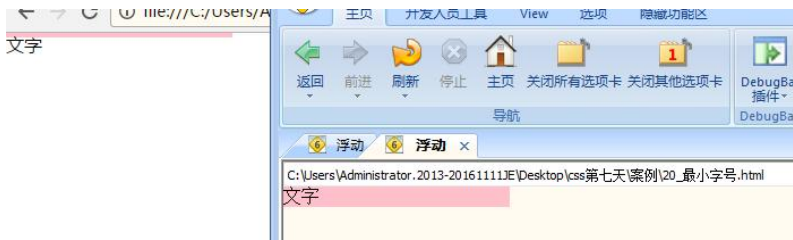
#### 5.2.2 最小高度

如果盒子高度小于默认字号，不会正常显示。高度会是最小的字号。

解决方法：

单独给 IE6 浏览器，强制给个很小的字号。

```
1      .box{
2          width: 200px;
3          height: 4px;
4          background: pink;
5          -font-size:0;
6      }
```



## 5.3 浮动的问题

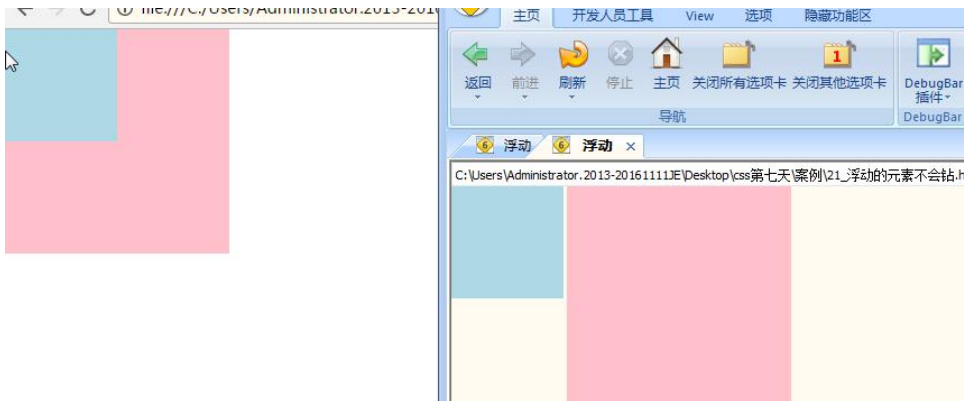
### 5.3.1 浮动的盒子不会钻到底下

情况：一个盒子浮动，一个盒子不浮动，在 IE6 里，不浮动的盒子不会钻到浮动盒子的下面占领它原来的标准流位置。

```

1  .box1{
2      width: 100px;
3      height: 100px;
4      background: lightblue;
5      float: left;
6  }
7  .box2{
8      width: 200px;
9      height: 200px;
10     background: pink;
11 }

```



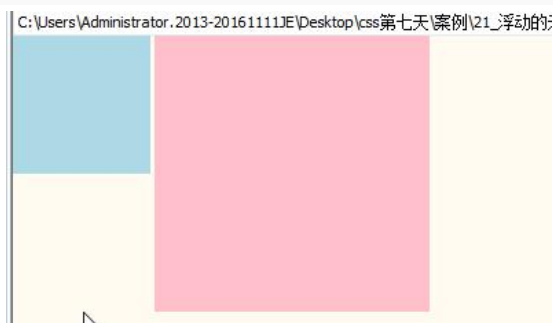
解决办法：同级的元素，要么都浮动要么都不浮动。

制作压盖效果必须用定位来实现。

### 5.3.2 3 像素的 bug

情况：一个浮动，一个不浮动，IE6 里两个盒子之间会出现 3px 的间距。

1



解决办法：给左边的盒子加负数的 margin-right

```

1  .box1{
2      width: 100px;
3      height: 100px;
4      background: lightblue;
5      float: left;
6      margin-right: -3px;
7  }

```

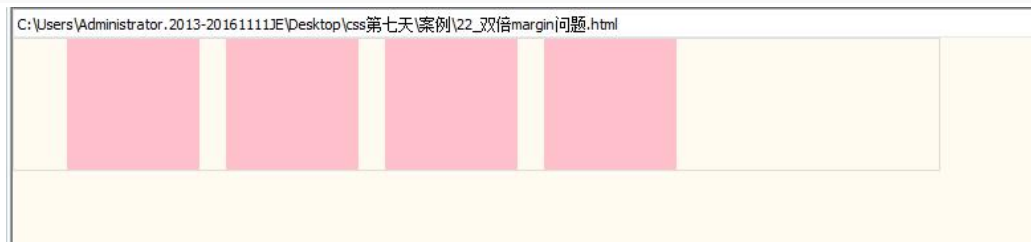
### 5.3.3 双倍 margin 问题

情况：一些元素浮动，有一个与浮动方向相同的方向的 margin，第一个元素会出现双倍边距的问题。

```

1  .box p{
2      float: left;
3      margin-left: 20px;
4      width: 100px;
5      height: 100px;
6      background: pink;
7  }

```



解决方法：

① 不允许父子之间的间距用儿子的 margin 去踹，用父亲的 padding 去挤出来，给第一个浮动的元素清除 margin，或者给反方向的 margin。

```

1  .box p{
2      float: left;
3      margin-right: 20px;
4      width: 100px;
5      height: 100px;
6      background: pink;
7  }

```

②（不提倡，不允许父子之间的间距用儿子的 margin 去踹）给第一个元素的 margin 减半。

```

1  .box p.first{
2      -margin-left: 10px;
3  }

```

### 5.3.3 overflow: hidden 失效

IE6 不认识 overflow: hidden; 引申作用，没法帮我们清浮动，自动撑高盒子。

IE 浏览器加载机制：

布局：layout

有布局：盒子本身加载的时候，根据自身内容来加载，会强制检测里面的元素，通过里面的元素性质进行自身布局。

没有布局：盒子本身加载的时候，根据祖先元素来加载，不会去管内部的元素。

IE 里有很多属性可以触发有布局的机制。触发 hasLayout 机制，就可以让 overflowhidden 生效。

有一个属性永远触发 hasLayout 机制。

```

1  _zoom: 1;

```

### 5.3 固定定位不支持

用 js 来模拟固定定位效果。

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1
1
1
1
1
1
1
1
1
1
1
1
1
1